



Lab - Rate Limiting

Wherein ...

- We also choose to add traffic throttling designed to mediate DDOS attacks, and
- Also want to manage how many requests individuals can make to our systems

Add a Spike Arrest policy

- Start Policy <Rate> @ "2pm"
 - 1 request every $60/2 = 30$ secs
- In Trace, Send repeatedly to Test for "500" error (validate policy is working)
- Reset <Rate> to "200ps"
 - 1 request every $1000/200 = 5$ msec

Spike Arrest policy

Add Step

Policy Instance

New

Existing

TRAFFIC MANAGEMENT

 Quota

 Spike Arrest

 Concurrent Rate Limit

 Response Cache

 Lookup Cache

 Populate Cache

 Invalidate Cache

 Reset Quota

Add a Quota policy

- Start Policy @ "2pm"
 - "Allow" = 2 ; "Interval" = 1 ;
"TimeUnit" = minute
- In trace, Test for "500" error code
- Reset "Allow" to "50"

Quota policy

Add Step

Policy Instance

New

Existing

TRAFFIC MANAGEMENT

 Quota

 Spike Arrest

 Concurrent Rate Limit

 Response Cache

 Lookup Cache

 Populate Cache

 Invalidate Cache

 Reset Quota

Quota Policy - Feed values from the API Product

- Find the API Product Quota variable names (in the docs or trace); Test

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Quota async="false" continueOnError="false" enabled="true" name="Quota-1" type="rollingwindow">
  <DisplayName>Quota-1</DisplayName>
  <Properties/>
  <Allow count="2" countRef="verifyapikey.Verify-API-Key-1.apiproduct.developer.quota.limit"/>
  <Interval ref="verifyapikey.Verify-API-Key-1.apiproduct.developer.quota.interval">1</Interval>
  <TimeUnit ref="verifyapikey.Verify-API-Key-1.apiproduct.developer.quota.timeunit">minute</TimeUnit>
  <Distributed>true</Distributed>
  <Synchronous>true</Synchronous>
  <StartTime>2016-6-1 12:00:00</StartTime>
</Quota>
```

THANK YOU